# 基于单根 I/O 虚拟化的密码设备中断频率优化方法 *

李　帅，孙　磊，郭松辉

(信息工程大学，郑州 450001)

**摘　要：**针对虚拟化环境下中断频率过大影响密码设备密码运算性能的问题，提出了一种降低中断频率的性能优化方法。首先建立了中断频率控制模型，通过实验验证了该模型的合理性和正确性；然后基于单根 I/O 虚拟化，在虚拟功能驱动层加入速度监测模块来实时监测加密速度的变化，并且当该模块监测到加密速度降低时自动调整虚拟功能中断频率上限，降低由中断频率过大带来的 I/O 传输消耗。实验结果表明，中断频率上限的调整显著提高了 I/O 密集型加密过程的加密速度。

**关键词：**中断频率；单根 I/O 虚拟化；控制模型；加密速度

**中图分类号：**TP302.7　　doi: 10.3969/j.issn.1001-3695.2018.03.0220

## Interrupt frequency optimization method of encryption device based on single root I/O virtualization

Li Shuai, Sun Lei, Guo Songhui

(*Information Engineering University*, *Zhengzhou* 450001, *China*)

**Abstract:** Aiming at the problem that the interrupt frequency in the virtual environment is too large to affect the cryptographic computing performance of the encryption device, this paper proposed a performance optimization method to reduce the interrupt frequency. Firstly, the method established the interrupt frequency control model and verified the rationality and correctness of the model through experiments. Then, based on a single root I/O virtualization, adding a speed monitoring module at the virtual function driving layer to monitor the change of the encryption speed in real time, and when the module detected that the encryption speed reduced, it automatically adjusted the upper limit of the virtual function interruption frequency to reduce the interruption frequency. The process reduced excessive I/O transfer consumption. The experimental results show that the adjustment of the upper limit of the interrupt frequency significantly improves the encryption speed of the I/O intensive encryption process.

**Key words:** interruption frequency; single root I/O virtualization; control model; encryption speed



## 0　引言

云计算是继分布式计算、网格计算、对等计算之后的一种新型计算模式错误!未找到引用源。，它以网络技术、虚拟化技术、分布式计算技术为基础，为云用户提供了强大的计算和存储能力。然而，伴随着云计算的快速发展，其面临的安全问题也日益凸显错误!未找到引用源。。2018 年，云安全联盟（cloud security alliance, CSA）最新公布的云计算面临的十二大顶级安全威胁对云安全提出了重大挑战。对此，各大云服务供应商（cloud service provider, CSP）也纷纷提出了各自的解决方案。例如，针对云中数据泄露的问题，阿里云将一种硬件安全模块（hardware security module，HSM）应用在底层硬件，通过虚拟化技术为上层应用提供数据加密服务错误!未找到引用源。。其中，HSM 本身是一种密码设备，也被称为加密加速器错误!未找到引用源。，可以确保安全的密钥管理，同时提供快速的加密操作。基于密码技术的云安全服务商——北京三未信安科技发展有限公司错误!未找到引用源。也提出了自己的云安全解决方案，该解决方案是在服务器端使用底层硬件设备——密码卡作为物理支撑，密码卡本身采用单根 I/O 虚拟化（single-root I/O virtualization, SR-IOV）错误!未找到引用源。技术实现云环境下的高性能资源共享。

云计算为用户提供了三种不同的服务模型，即软件即服务（software as a service, SaaS）、平台即服务（platform as a service, PaaS）以及基础设施即服务（infrastructure as a service, IaaS）。其最大的优点便是服务的高效性，且云服务的高效性主要体现

收稿日期：2018-03-19；修回日期：2018-04-18　　基金项目：国家重点研发计划项目（2016YFB0501900）

作者简介：李帅（1994-），男，河南驻马店人，硕士研究生，主要研究方向为 I/O 虚拟化、云计算安全（2650536170@qq.com）；孙磊（1973-），男，江苏靖江人，教授，博士，主要研究方向为云计算基础设施的可信增强和可信虚拟化；郭松辉（1979-），男，四川乐山人，副教授，博士，主要研究方向为云计算安全、虚拟化.

在云环境下的 I/O 传输性能上错误!未找到引用源。，所以云环境下的 I/O 性能对高性能计算机系统至关重要错误!未找到引用源。，大多数云服务提供商也采用了相应的 I/O 虚拟化技术为用户提供灵活高效的资源共享错误!未找到引用源。。其中，单根 I/O 虚拟化技术的出现显著提高了虚拟化条件下 I/O 传输性能，它采用 Passthrough I/O 错误!未找到引用源。传输方式，不需要虚拟机监控器（virtual machine monitor, VMM）对虚拟机（virtual machine, VM）进行监控，通过硬件直接与虚拟机进行 I/O 传输，并且在传输过程中，通过输入/输出内存管理单元（input/output memory management unit, IOMMU）减少了存储保护和地址转换的开销，大大提高了传输效率错误!未找到引用源。,错误!未找到引用源。。但是，当到达 SR-IOV 设备的外部中断请求过多错误!未找到引用源。时，会严重影响其 I/O 性能，I/O 性能的降低会直接影响 SR-IOV 密码设备的加密性能，这与高性能云计算的要求是背离的。

本文以 Intel Corporation DH895XCC Series QAT 错误!未找到引用源。密码设备为实验对象，采用 SR-IOV 技术在 KVM 平台上搭建了虚拟密码机（virtual cipher machine, VCM）集群，提出了一种通过监测加密速度变化来自动调节 VF 中断频率上限的控制方案，克服了大量虚拟中断带来的过高性能开销，利用 VF 驱动的可配置特点对 SR-IOV 密码设备在 KVM 中进行了实现，优化了大量虚拟中断造成的资源浪费，充分发挥了 SR-IOV 密码设备的性能。

## 1 相关工作和相关技术

### 1.1 相关工作

对于 SR-IOV 设备面临的由大量中断频率造成性能降低的挑战，学术界已开展了一些研究工作。综合国内外研究现状来看，SR-IOV 设备中断的研究主要从两个方面入手，第一个是直接从中断频率入手，通过降低中断频率来达到性能优化的目的；第二个是从中断机制入手，通过优化中断路径来达到性能优化的目的。对于直接减少大量中断频率来提高 I/O 性能的方法，Dong 等人错误!未找到引用源。指出了 SR-IOV 中断引起的性能降低问题，并在此基础上提出一种粗粒度中断频率控制方法错误!未找到引用源。对 XEN 中的虚拟中断处理进行了优化；但是该方法需要事先确定 VF 的最优中断频率上限，属于一种静态配置，不具有普适性。Guan 等人错误!未找到引用源。提出了一种基于事件的轮询模型，消除了一些关键 I/O 处理路径中引发的中断；但是该模型在主机内核需要 APIC 模拟，增加了主机的额外开销。Li 等人错误!未找到引用源。在粗粒度中断控制方法的基础上，提出了一种自适应中断频率控制的方法，该方法通过判断 SR-IOV 设备接收到的中断频率的大小与最优中断的关系，加入中断频率控制模块来控制中断频率从而提高虚拟机性能；但是该方法增加了 I/O 延迟，不适用于 I/O 密集型负载，而且仅是针对网卡虚拟化做了实现。对于中断机制的优化，学术界都是通过优化中断路径的方法来减少大量中断到来时 CPU 资源的过度消耗，从而提高 I/O 传输性能。Gordon 等人错误!未找到引用源。提出了一种更少中断退出的方法（exit-less interrupt, ELI），该机制采用软件模拟的方式在客户虚拟机内建立影子中断描述符表（interrupt description table, IDT）直接处理中断。但是在 ELI 机制中，影子 IDT 只能接收客户机已经分配的中断，其他没有被分配的中断仍需要主机的参与。而且，当有大量虚拟中断需要处理时，仍会导致较大的性能开销。Tu 等人错误!未找到引用源。提出了一种直接中断传输的方法（direct interrupt delivery，DID），该机制可在虚拟机中断处理过程中完全消除 VM-exit，优化了中断处理路径，且不需要软件模拟；但该机制是通过卸载与 root 模式有关的操作来禁止 VM-exit，存在潜在的安全问题，可能会导致虚拟中断的错误处理或者不处理。Hu 等人错误!未找到引用源。提出了一种针对 I/O 虚拟化的高效和相应事件系统（efficient and reSponsive event system for I/O virtualization, ES2），该系统改善了虚拟机与硬件设备之间的双向 I/O 传输，优化了 I/O 传输路径，可进行没有 VM-exit 的中断传输；但是该方法不适用于 SR-IOV 设备，SR-IOV 环境下中断的处理仍有主机的介入，当大量虚拟中断请求需要处理时，依然会造成频繁的上下文切换，严重影响系统性能。王展等人错误!未找到引用源。提出了一种基于单根 I/O 虚拟化的多根 I/O 资源池化方法，该方法实现了多个服务器对同一 I/O 设备的共享复用，减少了单体服务器连线的冗余，提高了资源利用效率；但是却造成了由中断导致的 I/O 设备资源竞争，当系统中存在大量中断请求时，会导致服务器 I/O 性能严重下降。

针对虚拟化环境下中断频率过大影响 SR-IOV 密码设备密码运算性能的问题，本文主要完成了以下工作：

a）模型建立。本文以 AES 算法为例分析了加密速度、加密字节块和中断频率之间的关系，并建立了相应的数学模型，理论推导出了三者之间的关系。结果表明，影响虚拟密码机加密性能的主要因素是中断频率的大小。

b）中断频率优化。本文提出了一种通过监测加密速度变化来自动调节虚拟功能（virtual function, VF）错误!未找到引用源。中断频率上限的控制方案。该方案在 VF 驱动层加入速度监测模块，可实时监测加密速度的变化，并且在整个监测的过程中，速度监测模块只需调整一次 VF 的中断频率上限，而且能使中断频率上限调整后的虚拟密码机加密性能达到最大。

c）实验验证。本文对得出的结论进行了实验验证，结果表明优化后的模型能最大限度地保持虚拟密码机加密性能。

### 1.2 相关技术

SR-IOV 协议是由 PCI-SIG 组织发布的 PCIe 总线互连协议的扩展，它通过 I/O 设备自身的硬件虚拟化将单个物理设备呈现为一个物理功能（physical function, PF）和若干虚拟功能错误!未找到引用源。。其中，每个 PF 都具有标准的 PCIe 功能，可以对其进行完全的配置管理。用户可以通过 PF 配置或者控制 PCIe 设备，也可以通过 PF 对数据的输入和输出进行管理。相比于 PF，VF 仅具有轻量级的 PCIe 功能，能进行数据的传输错误!未找到引用源。。在数据传输过程中，由于每个 VF 对应唯一的资源标志符（resource identifier ,RID），且每个 RID 可用于索引 IOMMU 页表，所以

每个 VF 可以独立接收和传输数据包。并且每个 VF 都拥有与性能相关的资源，如传输和接收描述符，同时可共享其他主要设备资源，实现了资源的高效共享。

目前，硬件密码设备技术比较有代表性的是 Intel 公司的 Intel Corporation DH895XCC Series QAT 密码设备。QAT 密码设备借助英特尔 QuickAssist 技术（英特尔® QAT），可提升云、网络、大数据和存储应用中动态数据和静态数据的安全性和压缩性能，同时可加快计算密集型操作，为安全性、身份验证和压缩提供了软件化的基础，显著提高了标准平台解决方案的性能和效率错误!未找到引用源。。该设备集成了对称加密和身份验证、不对称加密、数字签名、公开密钥加密和 DH 等加密技术，用户可以通过相应的接口调用相应的密码算法来实现加密需求。为了实现高性能云服务的需求，QAT 密码设备本身支持 SR-IOV 技术，通过 SR-IOV 技术的实现来达到云环境下资源的高效共享。其基于 SR-IOV 技术的 QAT 密码设备基本架构如图 1 所示。其中，VCM 与 VF 之间通过 VFIO 驱动直接通信，并且不受 VMM 的干预。

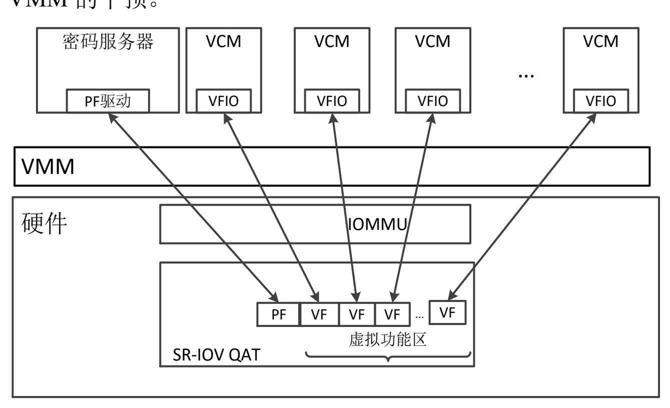



图 1 基于 SR-IOV 的 QAT 密码设备基本架构

## 2 关键问题分析

为保护云中数据的安全性，在云资源中心服务器集群中使用硬件密码设备为上层应用提供相应密码服务，采用 SR-IOV 技术将单个硬件密码设备虚拟成多个虚拟密码设备，每个虚拟密码设备分配给一个虚拟机当作虚拟密码机来使用，从而实现单个硬件物理资源的高效利用。用户通过租用虚拟密码机获得相应密码服务，当虚拟密码机处理用户发出的密码运算请求时，任务的中断会使当前执行密码运算的速度严重下降，而且，当云环境下存在大量的中断请求时，虚拟密码机的性能将会严重下降。

图 2 的测试显示了虚拟密码机处理单个密码任务时，对接收到的不同加密字节块大小，AES128 算法加密速度的变化趋势。从实验结果中可以看出，加密字节在 16 384 之前，虚拟密码机加密速度一直在增大；当加密字节为 16 384 时，虚拟密码机加密速度达到最大；当加密字节再继续增大时，虚拟密码机加密速度开始降低。整个过程中，随着加密字节块的增大，中断频率一直在增大，这说明当加密字节达到 16 384 字节时，SR-IOV 设备达到其最大处理中断请求和密码运算的能力；之后，中断频率的增加只能导致性能的下降，这是因为大量的中断请求占用了大量的 CPU 资源，致使 CPU 可用于处理密码运算的时间急剧减小。

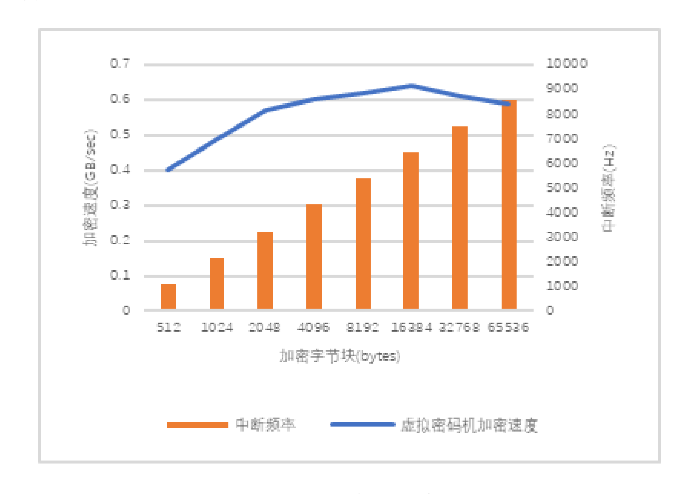



图 2 AES128 加密速度测试

图 3 的测试显示了虚拟密码机同时处理多个相同密码任务时，每个加密任务的加密速度变化以及 CPU 的使用率情况。从实验结果中可以看出，当虚拟密码机同时处理两个相同密码任务时，每个密码任务的加密速度大约是虚拟密码机处理单个密码任务时加密速度的一半；当虚拟密码机同时处理三个相同密码任务时，每个密码任务的加密速度大约是虚拟密码机处理单个密码任务时加密速度的三分之一，以此类推。该测试结果说明，当一个虚拟密码机同时处理多个密码任务时，由于 CPU 负载均衡的限制，每个负载平均分配可用的 CPU 资源，导致处理每个密码任务的速度均严重下降。但是虚拟密码机处理多个密码任务时，各个密码任务加密速度的总和小于虚拟密码机处理单个密码任务时加密速度的大小，而且每个虚拟密码机同时处理的密码任务越多，其加密速度的总和与单个密码任务的加密速度差别越大，这也从侧面反映出中断频率对虚拟密码机加密速度的影响：由多个任务带来的过多中断请求引起的虚拟密码机加密性能降低。

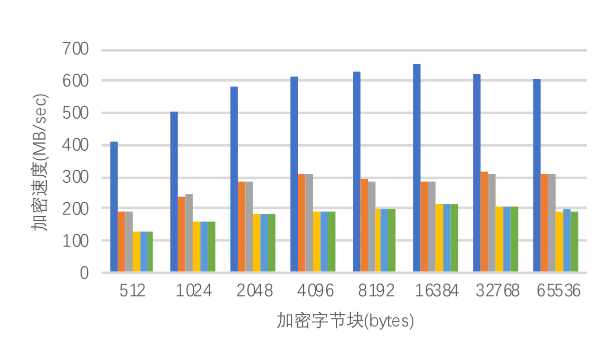



图 3 多个相同密码任务加密速度测试

因此，SR-IOV 密码设备仍面临着严峻的挑战：大量的中断请求频率导致 SR-IOV 密码设备性能降低。针对此挑战，本文探究了影响虚拟密码机加密性能的影响因素，建立了相应的数学模型，通过实验验证了理论分析结果，并在此基础上对不

同类型加密方式的差异性进行了分析和实验验证。

## 3　模型建立与优化

目前流行的加密和数字认证算法都是采用块加密方式，即把需要加密的明文分成固定大小的数据块，然后对其执行密码算法得到密文。本文以 AES 算法采用密码块链模式（cipher block chaining, CBC）错误!未找到引用源。加密为例，分析了影响 SR-IOV 密码设备加密性能的因素，并以加密速度作为性能参考标准，得出中断请求频率过大影响 SR-IOV 密码设备加密性能下降的结论。

### 3.1　中断频率控制模型的建立

假设 1 s 时间内，到达 SR-IOV 密码设备的中断请求频率为 $I$，SR-IOV 密码设备处理每个中断所用的时间为 $T$（其中 $T$ 为常数），执行密码运算所用的总时间为 $t_0$（$t_0<1$），1 s 时间内总的加密字节共分为 $n$ 组，每加密一组数据所用的时间为 $t$（其中 $t$ 为常数），则有

$$TI+t_0\le 1 \tag{1}$$

$$t_0=nt \tag{2}$$

假设每次向 SR-IOV 密码设备发送的加密数据包大小为 $B$，SR-IOV 密码设备对接收到的数据包进行加密的加密速度为 $V$（$V$ 由实验测得），分组大小为 $L$ 字节，对于特定的密码算法，其分组大小由 SR-IOV 设备本身决定，那么 $n$ 可表示为

$$n=\frac{1024V}{B/L}=\frac{1024VL}{B} \tag{3}$$

由式（1）~（3）可得

$$I\le\frac{1}{T}-\frac{1024VLt}{T}\frac{1}{B} \tag{4}$$

上述所有符号的单位及含义如表 1 所示。

表 1　每个符号的含义及单位

| 符号 | 含义 | 单位 |
|---|---|---|
| $I$ | 中断请求频率 | 一次 |
| $T$ | 处理每次中断所用的时间 | s |
| $t$ | 加密每组数据所用的时间 | s |
| $t_0$ | 执行密码运算所用的总时间 | s |
| $n$ | 总的分组数 | 组 |
| $B$ | 加密数据包大小 | Byte |
| $V$ | QAT 密码设备加密速度 | MB/s |
| $L$ | 分组大小 | Byte |

当式（4）等号成立时，对其求导，得出的结果如式（5）所示。

$$I^{(1)}=\frac{1024VLt}{T}\frac{1}{B^2} \tag{5}$$

从式（5）可以看出，$I^{(1)}$ 大于零，即 SR-IOV 密码设备的中断请求频率 $I$ 随着加密数据包 $B$ 的增大而增大，其中 $I^{(1)}$ 随 $B$ 的大致变化趋势如图 4 所示；相应的，中断请求频率 $I$ 随加密数据包 $B$ 的大致变化趋势如图 5 所示。

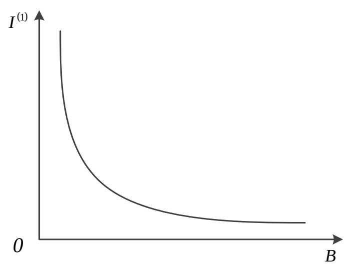



图 4　$I^{(1)}$ 随 $B$ 的大致变化趋势

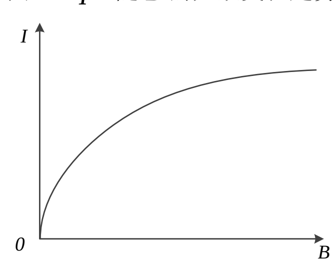



图 5　$I$ 随 $B$ 的大致变化趋势

即加密数据包 $B$ 越大，完成加密运算需要的运算次数越多，相应的中断请求的频率就会越大。

同时，由式（1）~（3）可得，加密速度 $V$ 与加密数据包 $B$ 和中断频率 $I$ 的关系如式（6）所示。

$$V\le\frac{B}{1024Lt}-\frac{BT}{1024L}I \tag{6}$$

式（6）取等号时，分别对加密数据包 $B$ 和中断频率 $I$ 求偏导，得出如下关系：

$$\begin{cases}\dfrac{\partial V}{\partial B}=\dfrac{1}{1024Lt}(1-TI)\\ \dfrac{\partial V}{\partial I}=-\dfrac{T}{1024L}B\end{cases} \tag{7}$$

显然，$\frac{\partial V}{\partial B}$ 大于零，$\frac{\partial V}{\partial I}$ 小于零。也就是说，加密速度 $V$ 随着加密数据包 $B$ 的增大而增大，加密速度 $V$ 随着中断频率 $I$ 的增大而减小。由于当加密数据包 $B$ 为 0 时，即没有中断请求时，加密速度 $V$ 为 0，所以加密速度 $V$ 一定存在极大值 $V_{max}$。当加密速度 $V$ 达到最大时，此时的加密数据包 $B$ 和 SR-IOV 密码设备处理的中断频率 $I$ 达到一个理想值，即在这个理想值之前，加密数据包 $B$ 和中断频率 $I$ 呈现正相关的关系，在这个理想值之后则呈现负相关的关系。这个理想值就是 SR-IOV 密码设备处理密码运算的最优值。若把加密数据包 $B$ 的理想值记为 $B_0$，在理想值 $B_0$ 下，SR-IOV 密码设备处理中断请求频率的理想值记为 $I_0$，则有

$$TI_0+\frac{1024V_{max}L}{B_0}t=1 \tag{8}$$

考虑极端的情况，令 $\frac{\partial V}{\partial B}$ 等于 0，求得驻点 $I=\frac{1}{T}$，显然，当 $I=\frac{1}{T}$ 时，加密速度为 0，此时的中断频率 $I$ 即 SR-IOV 设备每秒所能处理的最大中断请求，由于此时的加密数据包 $B$ 过大导致中断请求频率过大，致使 SR-IOV 密码设备把全部的时间用来处理中断请求，而失去了密码运算的能力。当然，这种现象在实验中是不会出现的。因为对于 SR-IOV 密码设备来说，其虚拟功能 VF 自身支持设置中断频率上限，为了追求更高的加密速度，中断频率的上限设置理论上应等于理想值 $I_0$，而理想值 $I_0$ 的取值需要通过实验测试获得数据。从理论分析的角度

可以得出结论：当加密数据包 $B$ 大于理想值 $B_0$ 时，中断请求频率 $I$ 将继续增大，但是由于已经超出了 SR-IOV 设备的密码运算能力，加密速度 $V$ 将会下降，甚至为 0，此时便会严重影响 SR-IOV 设备的密码运算性能。加密速度 $V$ 和加密数据包 $B$ 两者的大致变化趋势如图 6 所示。

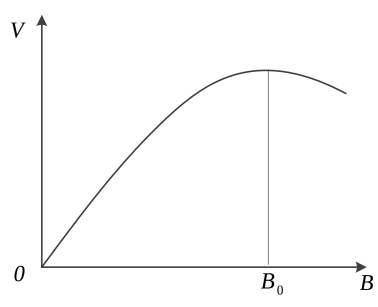



图 6 $V$ 和 $B$ 三者大致变化趋势

因此，有必要对中断请求处理的加密数据包大小进行控制，从而控制到达 SR-IOV 设备的中断请求频率，使 SR-IOV 设备处理密码运算的性能达到最大，即加密速度达到最大，来为用户提供更高效的密码运算服务。

### 3.2 中断频率优化方法

由上述分析可知，当加密数据包达到理想值 $B_0$ 时，此时 SR-IOV 密码设备加密速度最大；相应的，SR-IOV 密码设备处理的中断请求频率达到理想值 $I_0$。因此，可以对 SR-IOV 密码设备接收到的加密数据包的大小进行有效的控制来使加密速度尽量维持在一个较高的值。现在可作出初步设想：当 SR-IOV 密码设备接收到的加密数据包大小小于理想值时，不作任何改变；当 SR-IOV 密码设备接收到的加密数据包大小大于理想值时，先对接收到的加密数据包进行分块，且分块时以理想值作为分块基准，这样便能使每块加密字节获得最大加密速度。对 SR-IOV 密码设备接收到的加密数据包进行控制的模型如式（9）所示。

$$B=\begin{cases}B & B<B_0\\ \left\lceil \dfrac{B}{B_0}\right\rceil B_0 & B\geq B_0\end{cases} \tag{9}$$

其中：$\left\lceil \dfrac{B}{B_0}\right\rceil$ 为整数，存在小数点时进 1，其含义是加密数据包分块数目。相应的，加密速度可表示为

$$V=\begin{cases}\dfrac{B}{1024Lt}-\dfrac{BT}{1024L}I & I<I_0,B<B_0\\ V_{max}=\dfrac{B_0}{1024Lt}-\dfrac{B_0T}{1024L}I_0 & B\geq B_0\end{cases} \tag{10}$$

其中：中断请求频率 $I$ 为 SR-IOV 密码设备每秒实际接收到的中断请求频率，在加密速度没有达到最大值之前，SR-IOV 密码设备接收到的中断请求频率必定小于理想值。然而实际上 SR-IOV 设备接收到的加密数据包的大小是由用户决定的，其设备本身只是为虚拟密码机提供密码运算服务，因此无法改变接收到的加密数据包大小，只能从中断请求频率入手来控制 SR-IOV 密码设备每秒所能处理的密码运算量。由式（8）可知，当加密速度取得最大值 $V_{max}$ 时，加密数据包和密码设备接收到的中断请求频率达到相应的理想值 $B_0$ 和 $I_0$。如果控制密码设备每秒接收到的中断请求频率，将每个 VF 的中断频率上限均设置为 $I_0$，那么即使 SR-IOV 密码设备每秒接收到的加密数据包大小大于理想值 $B_0$。由于中断频率的限制，SR-IOV 密码设备仅能恰好处理 $B_0$ 大小的加密数据包，而此时的加密速度正好达到最大值 $V_{max}$。

由于事先对 VF 的中断频率上限进行设置存在局限性，需要先通过实验测试获得数据，找到最优中断频率，再对 VF 的中断频率上限进行设置，而且考虑到不同密码算法复杂度不同，CPU 处理不同密码算法所消耗的时钟周期也不同，则达到某种密码算法的最大加密速度所对应的中断频率会存在差异，因此该方法不具备普适性。为了能让中断频率能根据加密速度的变化自动调节阈值，本文提出了一种通过监测加密速度变化来自动调节 VF 中断频率上限的控制方案。该方案的架构如图 7 所示。

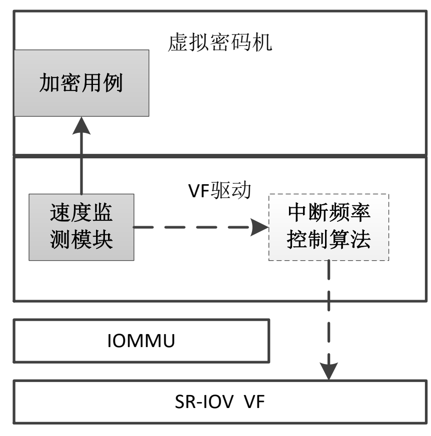



图 7 中断频率控制架构

该方案在 VF 驱动层加入速度监测模块，可实时监测加密速度的变化。当加密速度出现下降的趋势时，即下一秒的加密速度低于上一秒的加密速度时，速度监测模块通过中断频率控制算法自动将 VF 中断频率上限设置为上一秒时的 VF 中断频率，并且之后不作更改。对于 VF 中断频率上限初始值的设置一般设为理想值，因为实际上 CPU 不可能把所有时间放在处理中断请求上，在实际实验时，最优中断频率取值绝不会超过理想值 $I_0$。在整个监测的过程中，速度监测模块只调整一次 VF 的中断频率上限，此时调整的中断频率上限必定小于初始值。

速度监测模块的中断频率控制算法实现如下所示：

**Algorithm** controlling of interrupt rate

**Initialization**: Interrupt Rate $I\leftarrow I_0$

**Handing**:

$t=0,i=0,j=0,k=0$

get the speed of the encryption $V$

$i\leftarrow V$

$j\leftarrow i$

$t\leftarrow(t+1)$

$i++$

$i \leftarrow V$

**if** $i > j$ **then**

$k \leftarrow$ get current Interrupt Rate( )

**else if** $i < j$ **then**

$I \leftarrow k$

**end if**

通过在速度监测模块中加入中断频率控制算法，VF 驱动便可实时监测虚拟密码机处理加密用例时加密速度的变化，当监测到虚拟密码机加密速度达到最大值时，便自动将 VF 中断频率上限调整为虚拟密码机最大加密速度时的中断频率，即最佳中断频率，从而保证 CPU 处理中断所占用的时钟周期不在继续增大，并能以最大处理能力处理虚拟密码机接收到的加密数据包，使虚拟密码机的加密速度始终保持在最高水平，充分提高虚拟密码机加密性能。

## 4　加密性能测试

为了体现虚拟密码机在引入中断频率控制算法后加密性能的提升，实验中分别对每一台虚拟密码机的 VF 驱动添加了速度监测模块，以加密速度的大小来判断虚拟密码机加密性能。判定标准为：加密速度越大，虚拟密码机加密性能越好；反之，加密速度越小，虚拟密码机加密性能越差。本节实验环境为：主机和虚拟密码机操作系统均为 CentOS-7-x86_64-1511，内核版本为 Linux 3.10.0-327.e17.x86_64，处理器为 Intel(R) Xeon(R) CPU E5-2620 v3 @2.40 GHz，CPU 核心数为 12 核，内存大小为 128 GB，支持 VT-x 技术，且支持 SR-IOV，每台虚拟密码机均分配 1 个 vCPU 和 2048MB 内存，密码设备为 Intel Corporation DH895XCC Series QAT。

本章分别选取了对称加密算法 AES128、杂凑算法 SHA256 和公钥算法 RSA 来代表不同类型密码算法，探究其加密性能提升的差异性。测试工具为 crytodev。Crytodev 是调用密码设备的一个 benchmark 工具，可以测试密码设备中密码算法的运算速度，其中包含 AES128、SHA256 和 RSA 等加密算法。

### 4.1　AES128 加密性能测试

如 2.2 节所述，当虚拟密码机需要处理的加密字节块增大时，中断请求频率也随之增大，但是加密速度增大到一定值后出现下降的趋势。由第 3 章的分析可知，过大的中断请求频率会占用较多的 CPU 时钟周期，导致 CPU 用于处理密码运算的时钟周期严重减少，从而影响虚拟密码机加密速度。在虚拟密码机 VF 驱动层加入速度监测模块后，AES128 加密速度和中断频率的变化如图 8 所示。由图 9 可知，当加密字节块达到 16 384 字节时，AES128 加密速度达到最大，相应的最优中断频率约为 6 450。加入速度监测模块后，中断频率不再上升，相应的 AES128 加密速度在加密字节块为 32 768 时轻微下降了 0.1 GB，而后又恢复最大速度运行。这是由于速度监测模块在调整 VF 中断频率上限时占用了一部分 CPU 资源。中断频率上限调整结束后，虚拟密码机将保持最大加密速度运行。

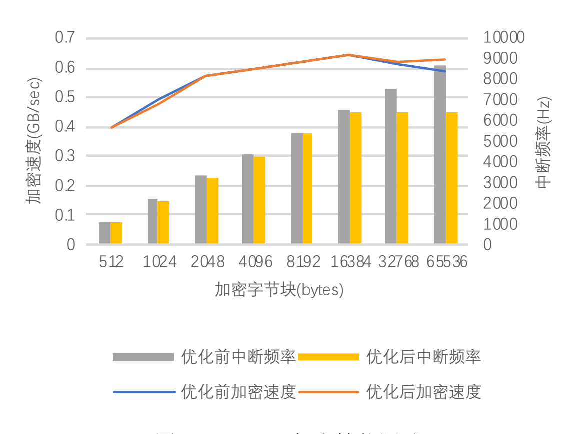



图 8　AES128 加密性能测试

### 4.2　SHA256 加密性能测试

与 AES 算法加密方式不同，SHA256 是一种杂凑算法，计算复杂度较高。当虚拟密码机对 SHA256 算法进行加密时，其加密速度的变化趋势与 AES128 算法大致相似，但最大加密速度和最优中断频率均不同。SHA256 算法的加密速度变化和中断频率变化如图 9 所示。由图 9 可知，当加密字节块为 16 384 字节时，SHA256 算法加密速度达到最大，相应的最优中断频率约为 5 109。加入速度监测模块后，中断频率不再上升，之后，虚拟密码机对 SHA256 算法的加密速度维持在 177 MB/s 左右。

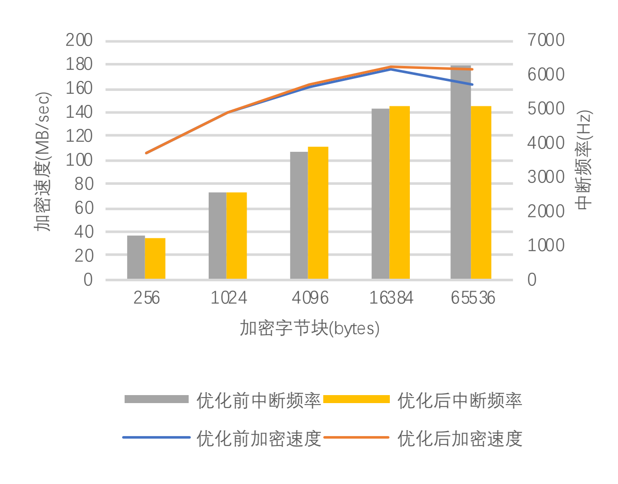



图 9　SHA256 加密性能测试

### 4.3　RSA 加密性能测试

对于 RSA 公钥密码算法来说，其硬件实现速度比较慢；相比于对称算法 DES 来说，RSA 硬件实现的速度比 DES 慢大约 1000 倍。实验中分别对 512、1 024、2 048 和 4 096 字节大小的数据块进行测试，发现字节越大，加密速度越慢，而相应的中断频率并不高。加入速度监测模块后，虽然 RSA 加密速度有一些提升，但是提升幅度不高，具体测试结果如图 10 所示。该测试结果表明，对于公钥算法 RSA 来说，本身复杂度比较高，其加密过程应归为计算密集型的操作，而中断频率的优化主要是针对 I/O 传输的优化，对于计算密集型操作的优化体现并不明显。反而对于对称算法 AES128 和杂凑算法 SHA256，其计算复杂度比公钥算法低得多，对于 I/O 传输的优化效果比较明显。

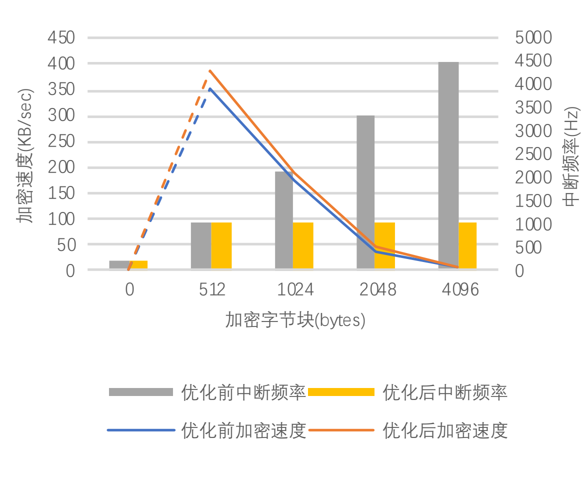



图 10　RSA 加密性能测试

## 5　结束语

为了解决 SR-IOV 密码设备的性能运算瓶颈问题，本文研究了影响 SR-IOV 密码设备密码运算性能的因素——过多的中断请求频率导致的 SR-IOV 密码设备加密性能降低，并建立了相应的中断频率控制模型，理论分析了中断频率、加密字节块和加密速度三者之间的关系，且在此基础上提出了一种中断频率优化方法。该方法是在 VF 驱动层添加了速度监测模块，并在模块内加入中断频率控制算法，可实时监测虚拟密码机加密速度的变化。当速度监测模块监测到虚拟密码机加密速度降低时，自动将 VF 中断频率上限调整为上一秒的中断频率，从而控制虚拟密码机接收到的中断请求频率，使 CPU 尽最大能力处理密码运算，充分提高了虚拟密码机加密性能。本文分别选用对称密码算法 AES128、杂凑算法 SHA256 和公钥密码算法 RSA 进行了实验验证。实验结果表明，对于 I/O 密集型加密过程，即使用对称密码算法加密和杂凑算法加密，在 VF 驱动层加入速度监测模块能充分发挥虚拟密码机加密性能；而对于计算密集型加密过程，即使用 RSA 算法加密，性能提升不明显。因此，本文方案对于 I/O 传输的优化起到了良好的促进作用，充分提高了虚拟密码机加密性能。

## 参考文献：


[1] 林闯，苏文博，孟坤，等. 云计算安全：架构、机制与模型评价 [J]. 计算机学报, 2013, 36 (9): 1765-1784. Lin Chuang, Su Wenbo, Meng Kun, *et al.* Cloud computing security: architecture, mechanism and model evaluation [J]. Journal of Computers, 2013, 36 (9): 1765-1784.

[2] 张玉清，王晓菲，刘雪峰，等. 云计算环境安全综述 [J]. 软件学报, 2016, 27 (6): 1328-1348. Zhang Yuqing, Wang Xiaofei, Liu Xuefeng, *et al.* An overview of cloud computing environment security [J]. Journal of Software, 2016, 27 (6): 1328-1348.

[3] 阿里云 [EB/OL]. https://www. aliyun. com/product/hsm?spm=5176. 8037491. 395145. 117. 438723976JJa2G.

[4] Koppel B, Neuhaus S. Analysis of a hardware security module's high-availability setting [J]. IEEE Security & Privacy, 2013, 11 (3): 77-80.

[5] 三未信安 [EB/OL]. http://ec. com. cn/html/2014/1016/17. html.

[6] Dong Yaozu, Yang Xiaowei, Li Jianhui, *et al.* High performance network virtualization with SR-IOV [J]. Journal of Parallel and Distributed Computing, 2012, 72 (11): 1471-1480.

[7] Akkinapalli K, Rao R R. A survey on encryption and improved virtualization security techniques for cloud infrastructure [J]. Global Journal of Computer Science & Technology, 2014, 14 (2) .

[8] Wang Guohui, Ng T S E. The impact of virtualization on network performance of Amazon EC2 data center [C]// Proc of the 29th Conference on Information Communications. Piscataway, NJ: IEEE Press, 2010: 1163-1171.

[9] Xu Xin, B. Davda. A hypervisor approach to enable live migration with passthrough SR-IOV network devices [J]. ACM Sigops Operating Systems Review, 2017, 51 (1): 15-23.

[10] Musleh M, Pai V, Walters J P, *et al.* Bridging the virtualization performance gap for HPC using SR-IOV for infiniBand [C]// Proc of IEEE International Conference on Cloud Computing. Washington DC: IEEE Computer Society, 2014: 627-635.

[11] Younge A J, Walters J P, Crago S P, *et al.* Supporting high performance molecular dynamics in virtualized clusters using IOMMU, SR-IOV, and GPUDirect [C]// Proc of ACM Sigplan//Sigops International Conference on Virtual Execution Environments. 2015: 31-38.

[12] Intel (R) quickAssist (QAT) crypto poll mode driver [EB/OL]. http://dpdk. org/doc/guides/cryptodevs/qat. html.

[13] Li Jian, Xue Shuai, Zhang Wang, *et al.* When I//O interrupt becomes system bottleneck: efficiency and scalability enhancement for SR-IOV network virtualization [J]. IEEE Trans on Cloud Computing, 2017, PP (99): 1-1.

[14] Dong Yaozu, Yang Xiaowei, Li Xiaoyong, *et al.* High performance network virtualization with SR-IOV [C]// Proc of IEEE International Symposium on High PERFORMANCE Computer Architecture. 2010: 1471-1480.

[15] Dong Yaozu, Xu Dongxiao, Zhang Yang, *et al.* Optimizing network I//O virtualization with efficient interrupt coalescing and virtual receive side scaling [C]// Proc of IEEE International Conference on CLUSTER Computing. Washington DC: IEEE Computer Society, 2011: 26-34.

[16] Gordon A, Amit N, Har'El N, *et al.* ELI: bare-metal performance for I//O virtualization [C]// Proc of Architectural Support for Programming Languages & Operating Systems. 2012: 411-422.

[17] Guan Haibing, Dong Yaozu, Kun Tian, *et al.* SR-IOV based network interrupt-free virtualization with event based polling [J]. IEEE Journal on Selected Areas in Communications, 2013, 31 (12): 2596-2609.

[18] Tu Chengchun, Ferdman M, Lee C T, *et al.* A comprehensive implementation and evaluation of direct interrupt delivery [C]// Proc of ACM Sigplan//Sigops International Conference on Virtual Execution Environments. 2015: 1-15.

[19] Hu Xiaokang, Zhang Wang, Li Jian, *et al.* ES2: Aiming at an optimal

virtual I//O event path [C]// Proc of IEEE International Conference on Parallel Processing. 2017: 141-150.

[20] 王展，曹政，刘小丽，等. 基于单根 I//O 虚拟化的多根 I//O 资源池化方法 [J]. 计算机研究与发展, 2015, 52 (1): 83-93. Wang Zhan, Cao Zheng, Liu Xiaoli, *et al.* A multiple I//O resource pooling method based on single root I//O virtualization [J]. Journal of Computer Research and Development, 2015, 52 (1): 83-93.

[21] Richter A, Herber C, Wallentowitz S, *et al.* A hardware//software approach for mitigating performance interference effects in virtualized environments using SR-IOV [C]// Proc of the 8th IEEEInternational Conference on Cloud Computing. 2015: 950-957.

[22] 英特尔® QuickAssist 技术 [EB/OL]. https://www. intel. cn/content/www/cn/zh/architecture-and-technology/intel-quick-assist-technology-overview. html.

[23] Frankel S, Glenn R, Kelly S. The AES-CBC cipher algorithm and its use with IPSec [S]. Ietf RFC, 2003.